



Solution Design Document



TABLE OF CONTENTS

I.	Purpose	3
II.	Automated process details	4
2	Runtime guide	5
2.1	<i>Architectural structure of the Master Project</i>	5
2.2	<i>Master Project Runtime Details</i>	5
2.3	<i>Project name</i>	6
2.4	<i>Project(s) workflows</i>	7
2.5	<i>Packages</i>	8
3	Other Details	8
	Future Improvements	8
	Other Remarks	Error! Bookmark not defined.
4	Glossary	10

I. PURPOSE

The application begins by processing user data from a centralized Excel file, organizing it into individual sheets for each country. It then retrieves real-time disaster alerts and awareness levels through API calls to the meteoalarm.org service, leveraging XML data for accuracy and up-to-date information.

Once the XML data is parsed, the application identifies which users are located in regions with active alerts. For each affected user, it generates a customized email that includes detailed information about the disaster alert, ensuring that critical information is communicated promptly and effectively.

The primary focus of this solution is to provide a robust, scalable, and efficient automation tool that minimizes manual intervention, reduces the potential for errors, and ensures reusability for future enhancements or additional functionalities. By automating this process, the application enhances disaster preparedness and communication, benefiting both end users and operational teams.

Other Remarks

- Workflow Scheduling: Ensure that the workflow is scheduled to run every night at 7 PM, but only after the daily report has been published to avoid sending outdated or incomplete information.
- Alert Data Accuracy: Regularly verify the reliability of the API data source (meteoalarm.org) to maintain trustworthiness.
- Email Customization: Ensure that email templates are user-friendly and accessible, with clear instructions for recipients to act on the alerts.
- Testing: Perform routine tests to confirm that all regions and alert types are correctly matched and processed.
- Support for Multiple Template Files: Implement functionality to handle different email templates based on the type or severity of the alert, or based on user preferences.
- User Feedback Integration: Introduce a feedback mechanism to track user response rates and improve email delivery strategies..

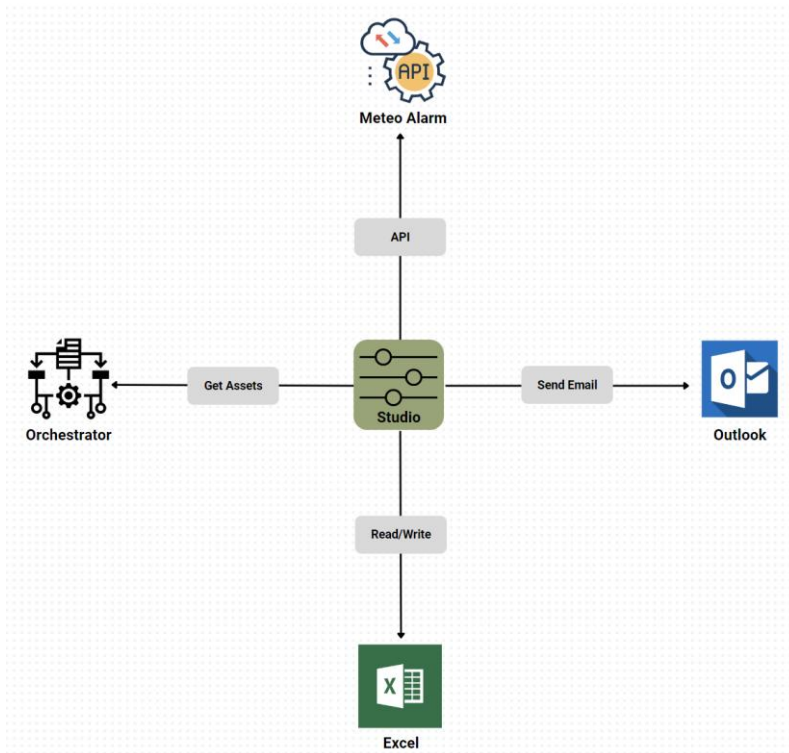
II. AUTOMATED PROCESS DETAILS

Details filled in need to reflect the actual information for the Master Project released for production. The following table will be populated:

Item	Description
Master Project Name	RPA_Meteo_Alert
Robot Type	BOR
Orchestrator used?	Yes
Scalable	Yes
UiPath version used	20.10.2

2 RUNTIME GUIDE

2.1 Architectural structure of the Master Project



2.2 Master Project Runtime Details

ITEM NAME	DESCRIPTION
Production environment details	<i>Fill in each bolded section - empty fields are not allowed. If the section does not apply to your automation then mark as n/a.</i> <i>The automation process runs on the local machine, manually started by the person operating it, and depends on API calls, website availability for retrieving weather alerts, Excel for input/output data management, and Outlook for sending email notifications, running until all entries are processed.</i>
Prerequisites to run	<i>n/a</i>
Input Data	<i>Excel file: Users.xaml</i>
Expected output	<i>Excel file Alerts.xaml and notification logs.</i> <i>The number of emails sent will match the number of alerts found for the requested regions specified by the recipients.</i>

How to start the automated process	<i>The process will be started manually by running the robot from UiPath Studio or Assistant on the local machine</i>
Reporting (queues reporting, Kibana or another platform)	<i>Example: Orchestrator logs and jobs dashboards.</i>
How is Orchestrator used?	<i>Orchestrator is used for securely managing credentials and dynamically providing file paths for various processes.</i>
Password policies (mention any specific compliance requests)	<i>n/a</i>
Stored credentials (Never use hardcoded credentials in the workflow!)	<i>Stored in Orchestrator Assets(credentials for API call)</i>
List of queues names (Naming convention: ProcessName_QueueName)	<i>n/a</i>
Schedule Details	<i>n/a</i>
Multiple Resolutions Supported? (in case of image automation / Citrix and VDI)	<i>N/A</i>
Recommended Resolution	<i>n/a</i>

2.3 Project name

RPA_Meteo_Alert Project

ITEM NAME	DESCRIPTION
	<i>Fill in each section - empty fields are not allowed. If the section does not apply to your automation then mark as n/a.</i>
Environment used for development (name, location, configuration details etc)	Laptop_ProjectEnv1 Local Development Machine Windows 11, Intel Core i7, 16GB RAM, 512GB SSD, UiPath Studio 2022.10, Excel 2019
Environment prerequisites (OS details, libraries, required apps)	<i>Windows 7, UiPath Studio , Microsoft Excel, Microsoft Outlook</i>
Repository for project (where is the developed project stored)	https://github.com/Stefan-Macovei-Grigoras/RPA-meteo-alarm
Configuration method (assets, excel file, Json file)	Json File, Orchestrator assets
List of reused components	<i>n/a</i>

List of new reusable components	<i>n/a</i>

2.4 Project(s) workflows

Workflows specific to: RPA_Meteo_alert

Workflow Name	Description
Main	<i>invokes all the other workflows</i> <i>Input Params : out_usersFilePath(String)</i> <i>out_alertsFilePath(String)</i> <i>out_meteoAlarmEndpoint(String)</i> <i>Output Params: in_usersFilePath(String)</i> <i>in_AlertsFilePath(String)</i> <i>in_meteoAlarmEndpoint(String)</i>
GetAssets	<i>Retrieves the file path for assets from the Orchestrator. Logs the start and throws an exception on timeout (6 seconds). Returns the file path as output.</i> <i>Input Params : none</i> <i>Output Params: out_usersFilePath(String)</i> <i>out_alertsFilePath(String)</i> <i>out_meteoAlarmEndpoint</i>
CheckUsersFileExistence	<i>Verifies if the users' file exists. Handles errors if the file is not found.</i> <i>Input Params : in_usersFilePath (String)</i> <i>Output Params:none</i>
ReadUsersCreateAlerts	<i>Reads the users' file, processes the data, and generates alerts in separate worksheets for each country.</i> <i>Input Params : in_usersFilePath(String)</i> <i>in_AlertsFilePath(String)</i> <i>Output Params: none</i>

APICalls	<i>Makes API calls using data from the alert file, retries on failure(2 times with 1 sec pause between), and updates the alerts excel file with the API response. Input Params : in_AlertsFilePath(String) in_meteoAlarmEndpoint(String) Output Params: none</i>
SendEmails	<i>Sends emails to users whose region has alerts, notifying them about the relevant information. Input Params : in_usersFilePath in_AlertsFilePath Output Params: none</i>

2.5 Packages

Package Name	Description
<i>No external packages used</i>	The automation uses only built-in UiPath activities, without external packages.

3 OTHER DETAILS

Future Improvements

Integration with Additional Data Sources

Incorporate weather data from other reliable services (e.g., OpenWeatherMap or NOAA) for enhanced alert coverage.

Support for multi-hazard warnings, such as earthquakes, floods, or wildfires, to provide a more comprehensive alert system.

Real-Time Notifications Across Multiple Channels

Extend notifications to SMS, mobile app push notifications, or voice calls for urgent alerts.

Allow users to select their preferred notification channel and customize alert preferences.

User-Friendly Interface for Customization

Develop a front-end dashboard for managing user data, setting preferences, and monitoring alert logs.

Include options for users to update their location and preferences directly through a web interface.

Localization and Multilingual Support

Add support for multiple languages to serve a diverse user base more effectively.

Geospatial Mapping Features

Visualize alerts on a map for users to better understand their proximity to affected zones.

Robustness

Includes data validation steps to ensure the integrity of user and alert information.

Efficiency

Automates the entire process, reducing the need for manual monitoring and data entry.

Optimized API calls and data parsing methods to minimize processing time.

Replicability

Clear architecture allows for straightforward duplication in other regions or scenarios.

Configurable components make it adaptable to different alert sources or user datasets.

Reusability of Components

Core components like the XML parser, email generator, and data processing modules are reusable in other applications.

API integration and error-handling frameworks can be repurposed for similar alerting systems.

4 GLOSSARY

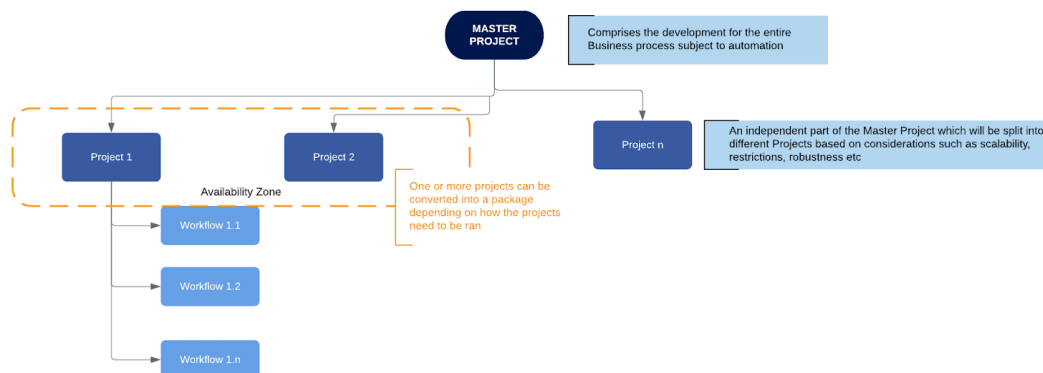
The main terms used in the Solution Architecture Document are defined below:

Master project - the overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation. There is a 1 to 1 connection between the Master Project and the Process to be automated (As presented in the PDD).

Project - an UiPath Studio project containing one or multiple workflow files. A project can be converted to a package and run independently, covering a particular scope within the master project. Or multiple projects can be converted into one package depending on the aims and restrictions of the automation. The project is used when defining the development and support phase of the automation.

Package - the output of compiling one or multiple projects. A package can be deployed on the robot machine and be executed by the robot service. Only one package can be executed at a given time by a robot. The package is used when defining the running phase of the automation.

Workflow - a component of the package, the workflow encapsulates a part of the project logic. The workflow can be of type: sequence, flowchart or state machine. A workflow is saved as an .xaml file inside the project folder. A workflow file can be invoked from another workflow and by default there is an initial workflow file that will run when executing the package.



Activity - an action that the robot executes.

Sequence - a workflow where activities are executed one after another, in a sequential order

Flowchart - a workflow where activities are connected by arrows and the logic of the workflow can be easily followed in a visual manner. The flowchart can also be exported as an image from UiPath studio.

State machine - a more advanced way of organizing a workflow, similar to a flowchart.

BOR - Back office robot

FOR – Front office robot

Orchestrator – Enterprise architecture server platform supporting: release management, centralized logging, reporting, auditing and monitoring tools, remote control, centralized scheduling, queue/robot workload management, assets management.